Hospital Management System

Presentation Created By: Saurabh Sapte

AIM OF THE PROJECT

➢ The establishment and improvement of **doctor-patient interaction system is a very important requirement**, especially now when the communication technology is developing rapidly.

➢ The advantages of web can be made full use of to make up the time and distance gap between **doctors and patients and to provide fast and adequate medical services.**

➢ Our Web Application helps health care organizations to manage their day to day operations, patients record, employee records, appointments and other administrative tasks.

➢ The platform, Web services and database technology are all gradually maturing, so that we can develop a doctor-patient interaction system on web application platform to meet the needs of the patient and **provide doctors more efficient and convenient means of communication with patients.**

➢ The goal of HMS Project is to simplifies the work of healthcare professionals and improve patient care by providing a centralized platform for managing all hospital-related task.

WHO IS YOUR END USER ?

➢ Admin:- Admin is the highest privileged user, who can Manage all Resources and Activities of Online Hospital Management System.

➢ Patient:- Patient can easily book appointment and able to access medical History. Also patient can register himself on website directly.

➢ Doctor:- Doctors can check scheduled Patients and also give prescription to their Patient. Also Doctor can select their weekely schedule.

➢ Receptionist:- Receptionist can admit and discharge the patient with concern of doctor. Also Schedule doctor to patients as per their primary symptoms mentioned by patient while booking appointment. Also Allocate Ward to Patient while admitting him.

➢ Accountant:- Accountant can generate Invoice (In pdf format) and Manage all transactions.

EXISTING SYSTEM

➢ In today's world if someone wants to book a Doctor's Appointment we need to call in clinic or personally go to that place and book the appointment.

➢ This consumes precious time of the patient. Also if the doctor cancels his / her schedule, the patient does not come to know about it unless he/she goes to the clinic.

DISADVANTAGES OF EXISTING SYSTEM

1. Lack of privacy

2. Risk in the management of the data.

3. Less Security

4. Low co-ordination between

5. Less User-friendly

6. Accuracy not guaranteed

7. Not in reach of distant users.

PROPOSED SYSTEM

➢ The patient will have to register into the application for the first time. On registering, the patient will receive a username and password.

➢ The patient will then **send a request for appointment. The Receptionist can either accept the appointment or reject** it on the basis of doctor availability. The database will get updated accordingly and the patient will get a confirmation message.

➢ The add-on to this system is that the patient will view the previous health history like assign doctor, prescription , if admitted then admit date and discharge date, medicines

➢ This will be very useful in case the patient tends to forget the appointment. Also doctor , receptionist, accountant can view patient history by using a unique ID.

➢ **Adding JWT Token based authentication to the proposed methods** is one possibility to eradicate security leaks.

ADVANTAGES OF PROPOSED SYSTEM

➢ Easy to use because all Details of patients , hospital administration etc. are quickly available 24 x 7 on application

➢ It can be easily accessed globally with help of Internet.

➢ Maintaining records will be easier because all details are stored in database and retrieved easily from it.

➢ Interactive and attractive design.

➢ Provides online appointment booking in hospital easily.

MODULE LIST

ADMIN

- Create Id & Password
- View All Employee List
- View All Patients List
- Add Employee
- Remove Employee
- Check and Updated Resources
- Add Resources

DOCTOR

- Login
- View Appointment List of Patients
- Add Medicine on the basis of diseases
- Add Prescription
- Select weekly schedule
- View Resources.

MODULE LIST

PATIENT

- ➢ Register
- ➢ Login
- ➢ Book Appointment
- ➢ View Appointment History
- ➢ View Previous Health History

RECEPTIONIST

- ➢ Login
- ➢ View Patient Appointment Details
- ➢ Assign Doctor to patient on the basis of weekly schedule of doctor
- ➢ Admit Patient
- ➢ Discharge Patient
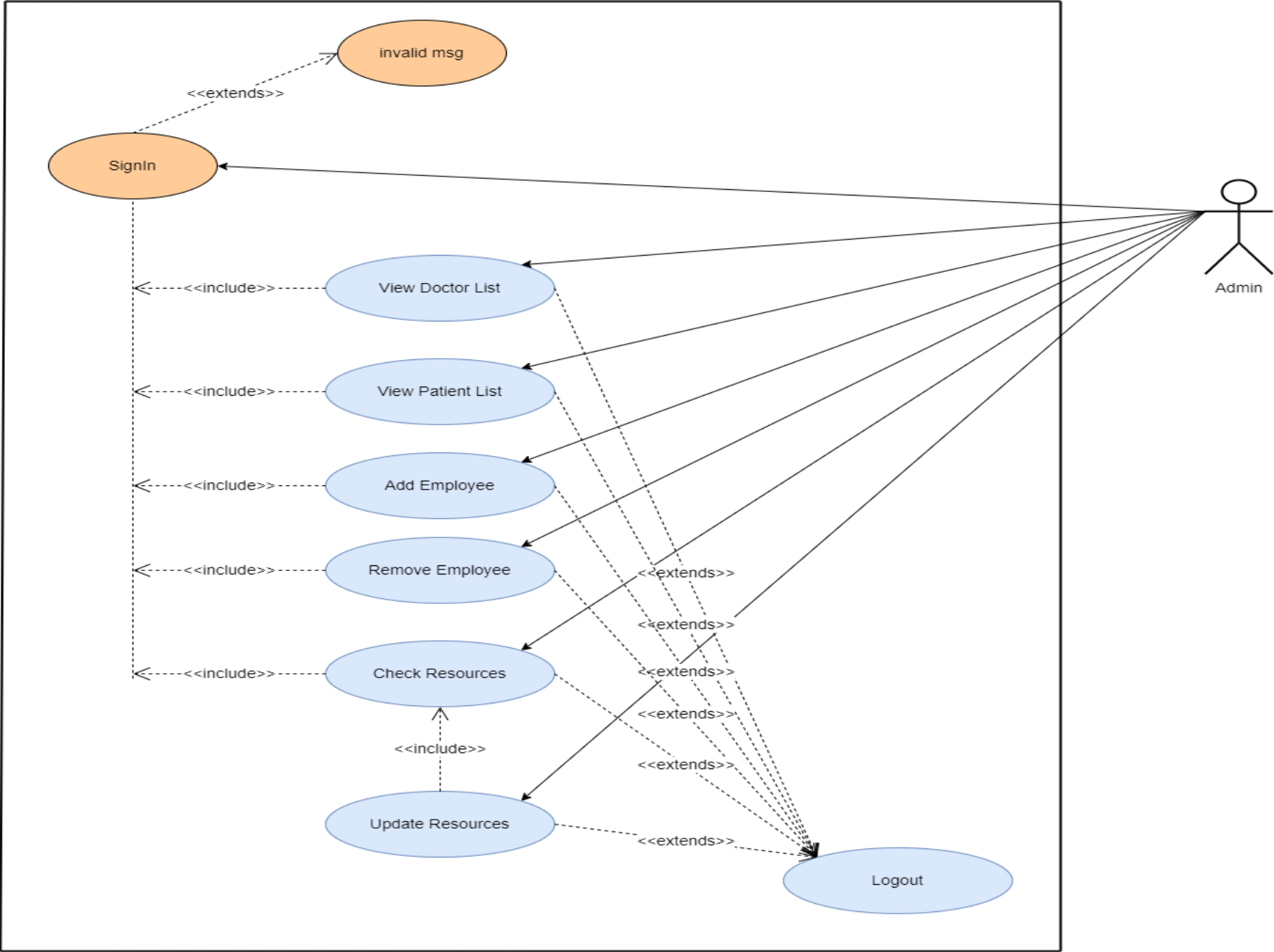- ➢ Update Patient Details
- ➢ Check Resources

CASHIER

- ➢ Login
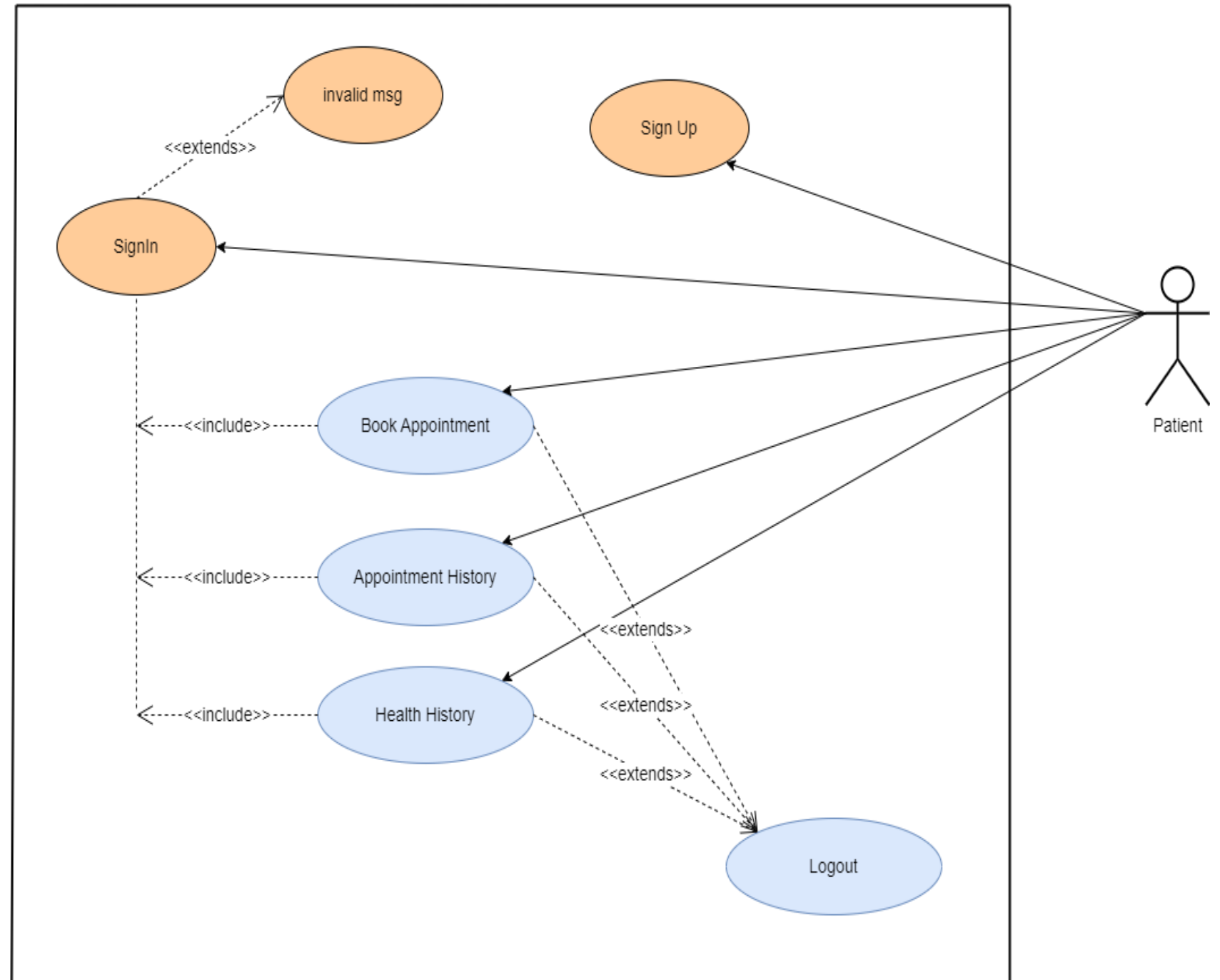- ➢ Payment Entry ,Update Payment Details and Generate Invoice (PDF)
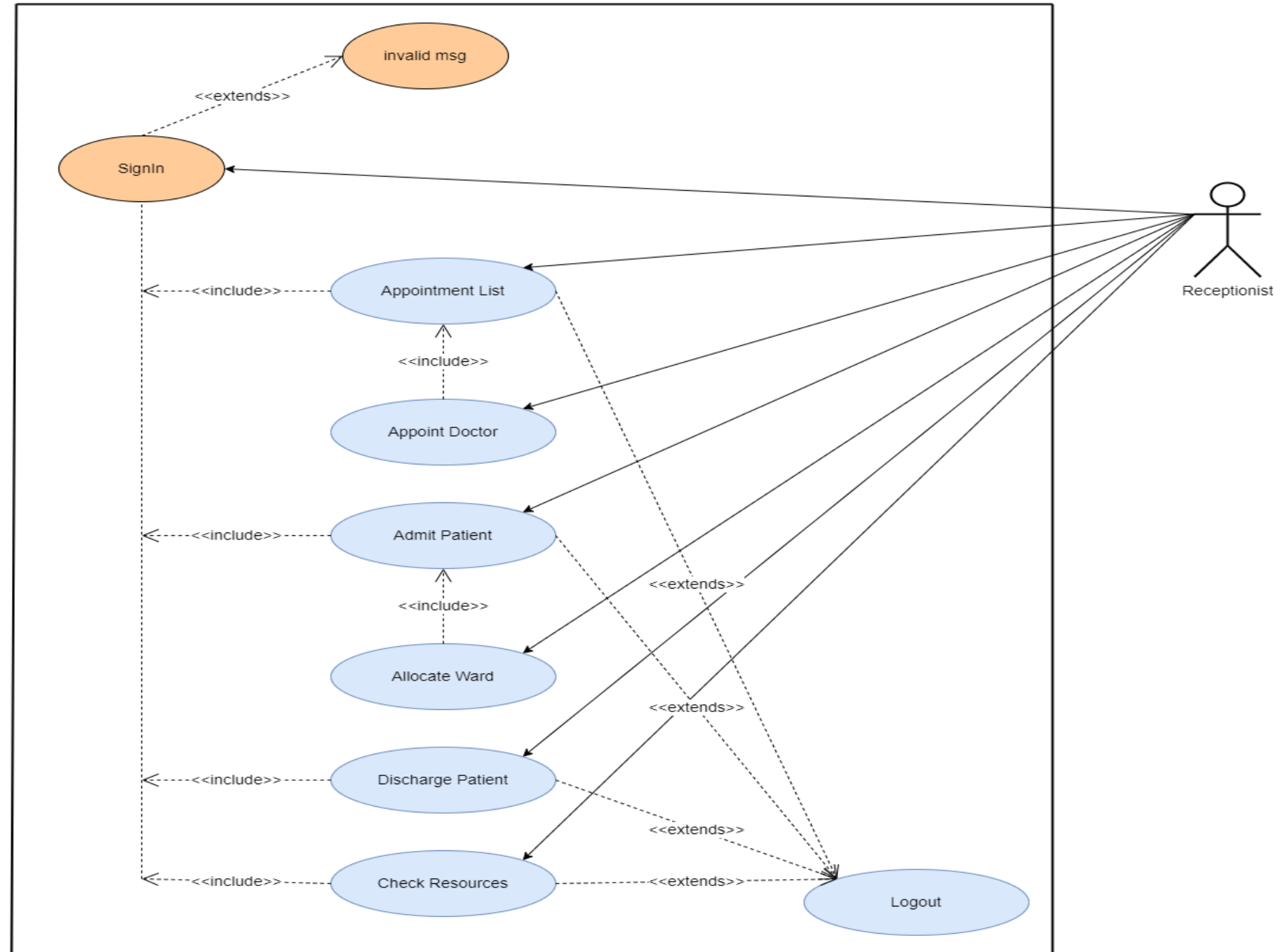
ER Diagram

USECASE DIAGRAMS

ADMIN

invalid msg

SignIn

<<extends>>

View Doctor List

<<include>>

View Patient List

<<include>>

Add Employee

<<include>>

Remove Employee

<<include>>

Check Resources

<<include>>

<<extends>>

<<extends>>

<<extends>>

<<extends>>

<<extends>>

Update Resources

<<include>>

<<extends>>

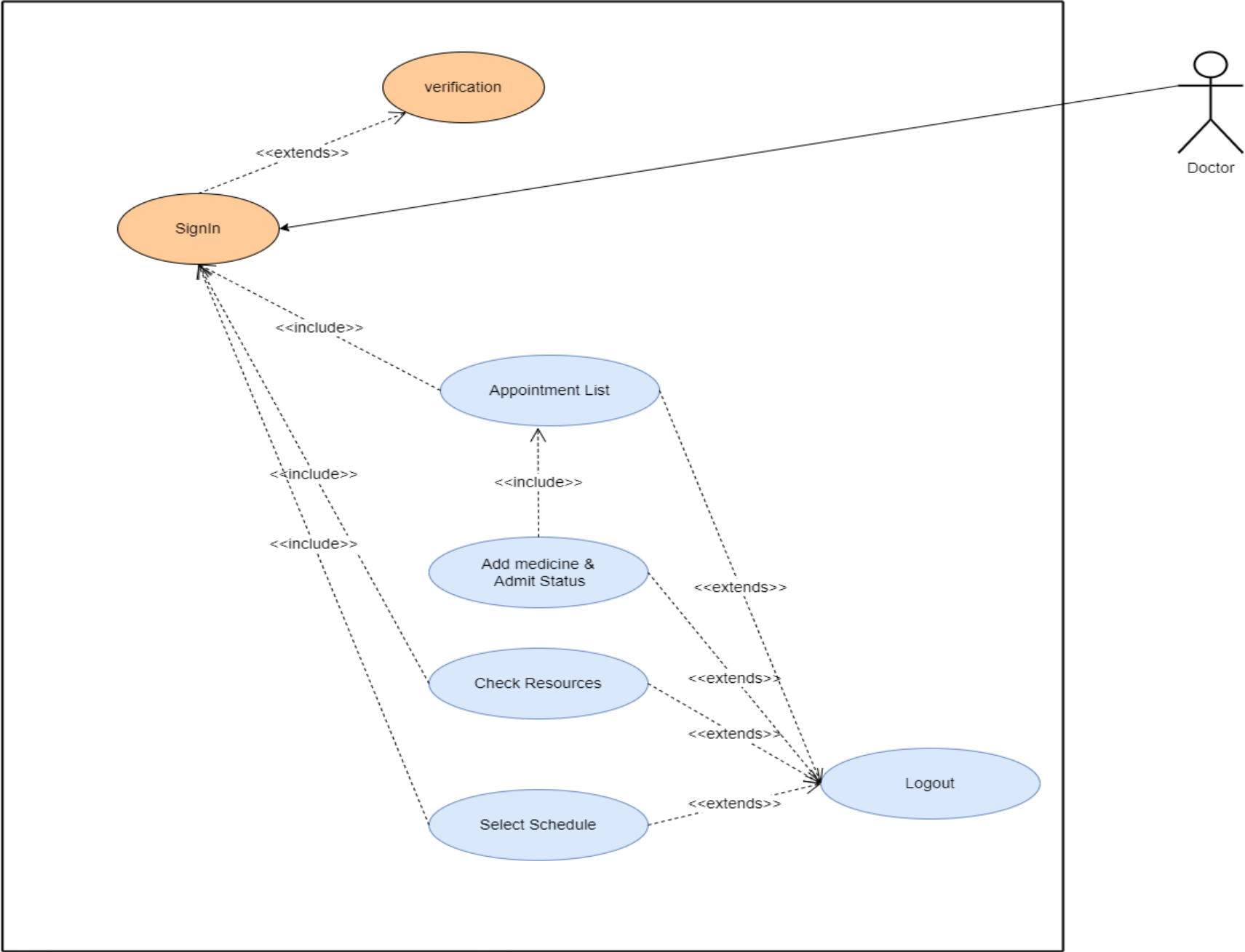Logout
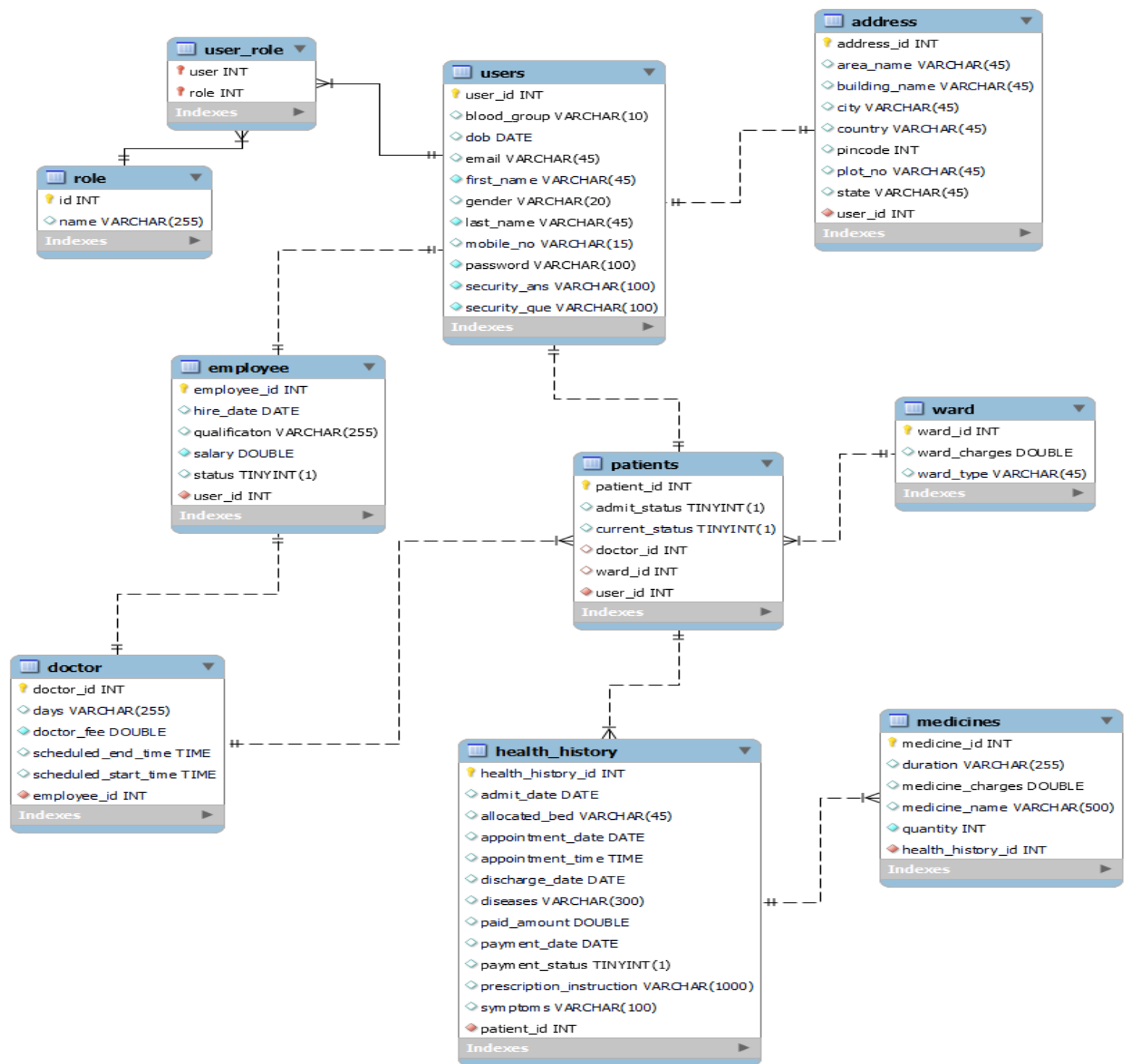
Admin

USECASE DIAGRAMS

USER/ PATIENT

USECASE DIAGRAMS
RECEPTIONIST

USECASE DIAGRAMS

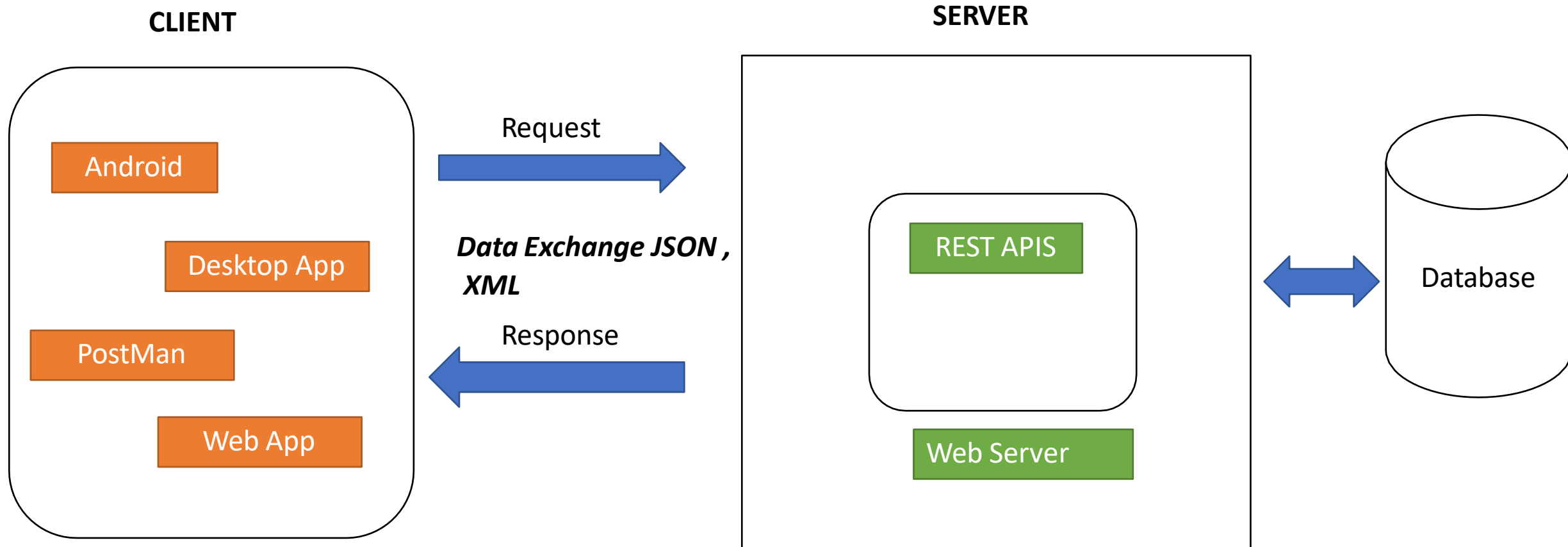## DOCTOR

USECASE DIAGRAMS
ACCOUNTANT

invalid msg

SignIn

Accountant

<<extends>>

<<include>>

Patient List

<<extends>>

<<include>>

Update Status

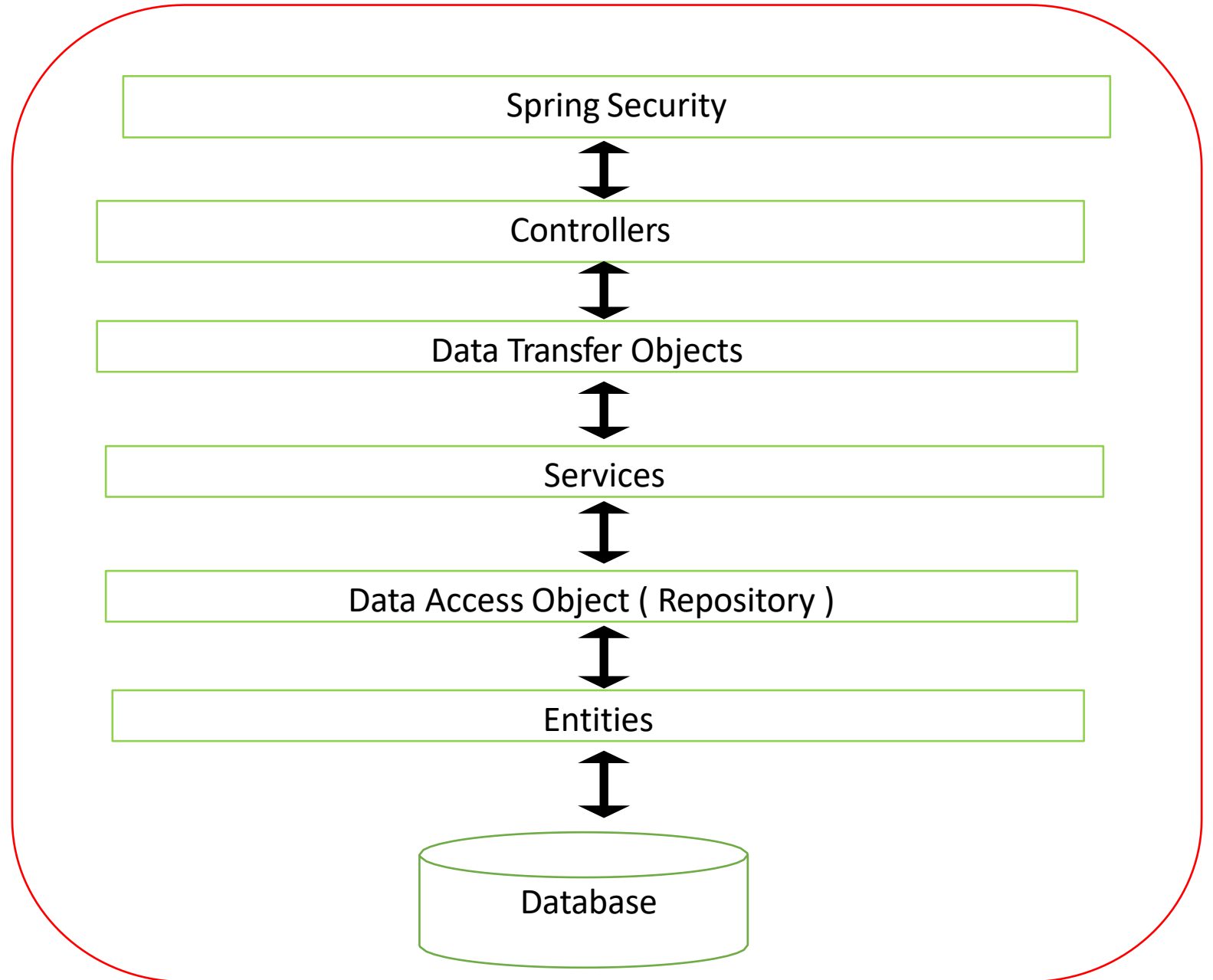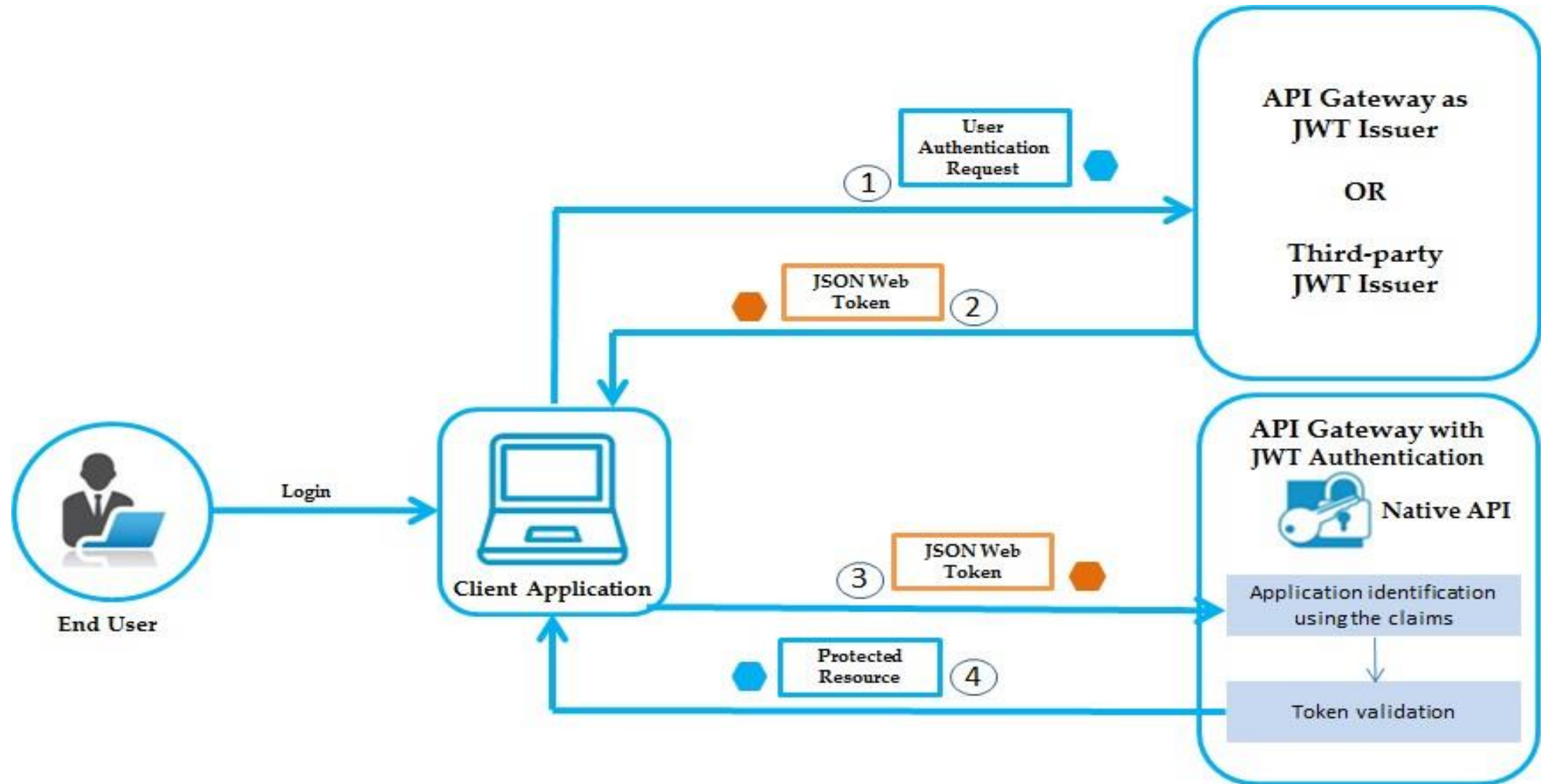<<extends>>

<<include>>

Generate Invoice

<<extends>>

Logout
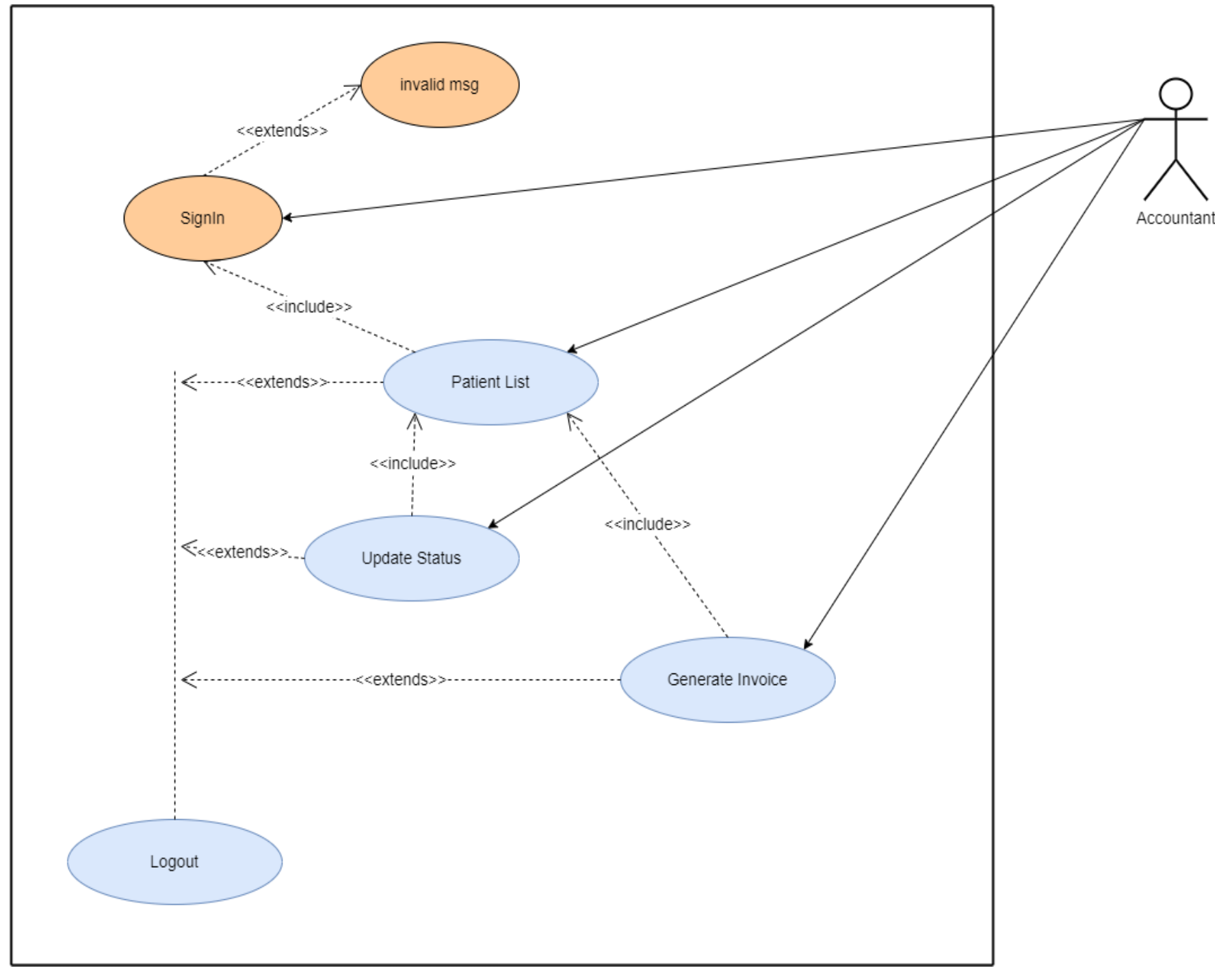
WORK FLOW DIAGRAM OF CLIENT SERVER ARCHITECTURE

SPRING TIER ARCHITECTURE DIAGRAM

JWT TOKEN BASED AUTHENTICATION

Password

Password

Token

Authorization Server

Password

User

Client

Token

Resource

Resource Server

MICROSERVICES
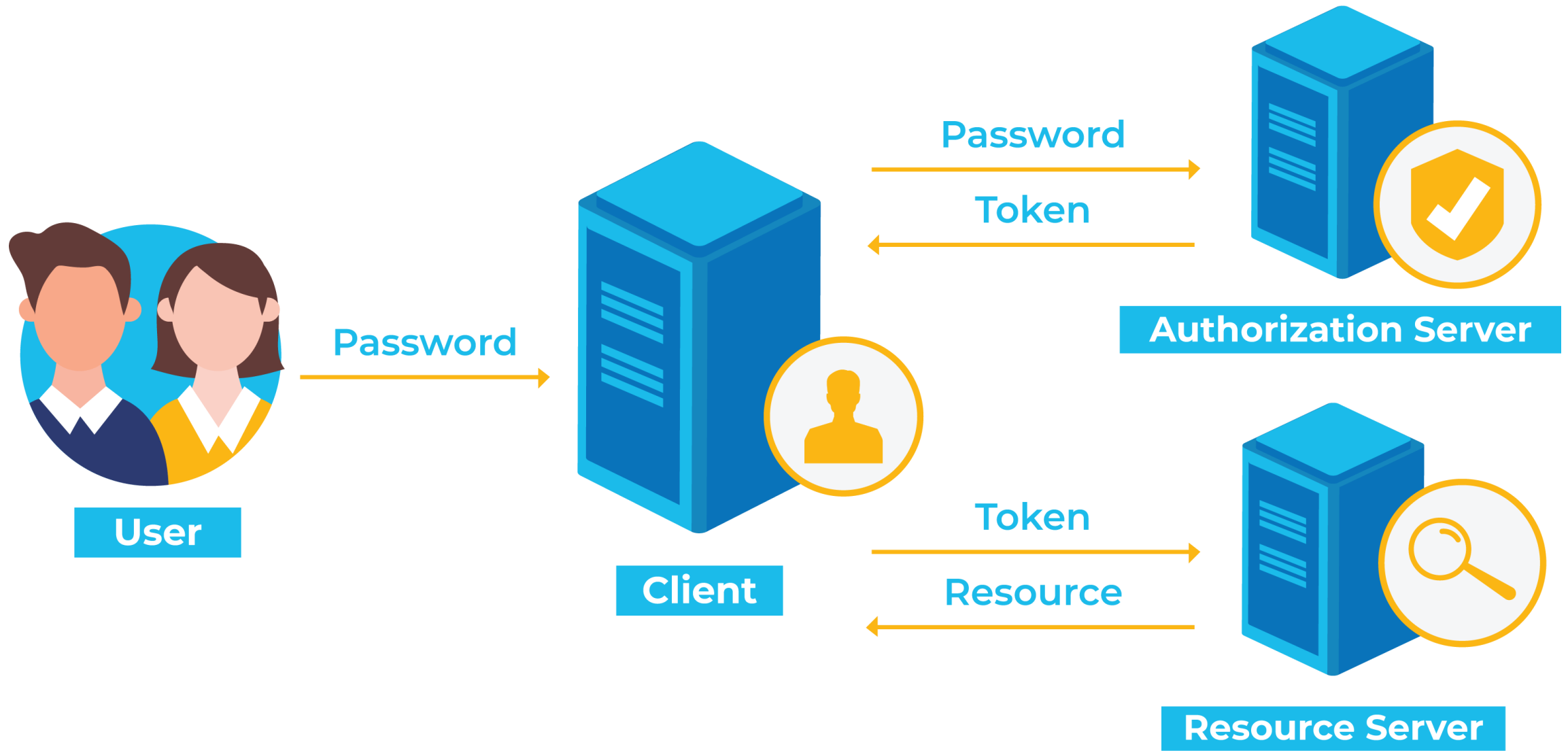
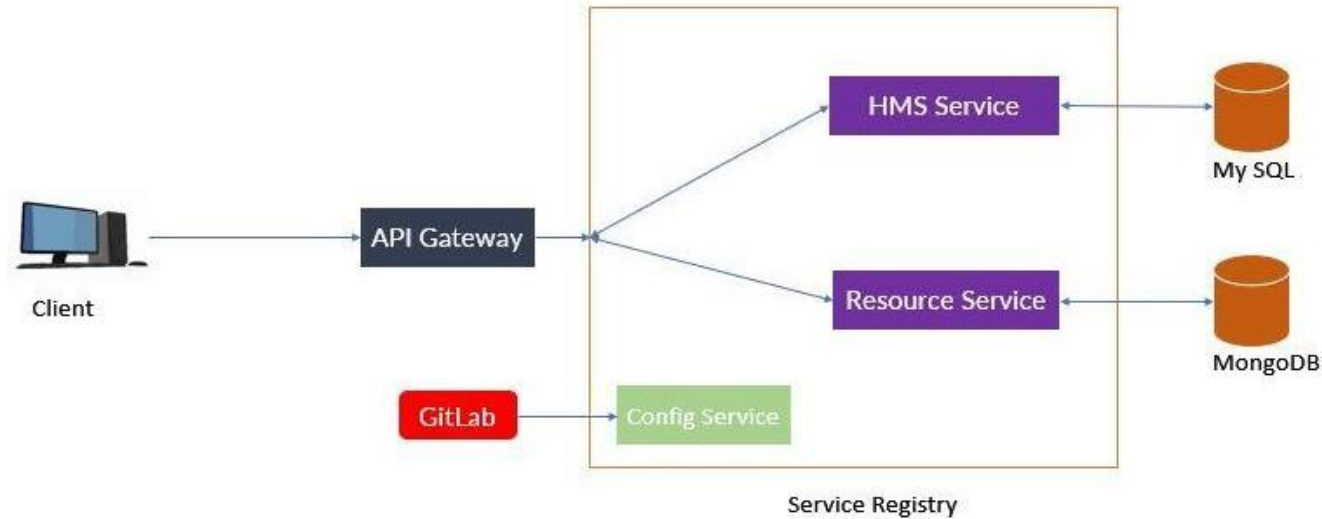**Why not monolithic architecture?**

➢ As project scale, it becomes difficult to manage.

➢ For single change redeployment of whole application needed.

➢ Difficult to adapt new technology for single functionality (affects whole application).

➢ Single bug may down your whole application.

MICROSERVICES

Dig. Microservices Architecture

- Microservices are smaller services that works together and also communicating with each other directly using light weight protocols like http.
- Microservices architecture breaks down application to independent standalone small applications.

Advantages
1) It is possible to change or upgrade each service individually rather than upgrading in the entire application.
2) One service may down without impacting to others.
3) Easily use different technologies for building different microservices. (Gives flexibility to choose technologies and framework for each microservice independently)
4) Less dependency. Loosely coupled.

Explanation
- Api Gateway :
  1) It provides single point to the client to which client will communicate and based on url send by frontend/client gateway decides which microservice need to be call. (we can say api gateway doing work of port forwarding based on request url by react as frontend)
  2) we have implement API gateway by using spring cloud gateway.
  3) It includes many build-in Route predicate Factories. All these predicates (in predicate there are different paths) match different attributes of the http request.

```
spring:
  application:
    name: API-GATEWAY
  cloud:
    gateway:
      routes:
      - id: HMS-SERVICE
        uri:
           lb://HMS-SERVICE
        predicates:
        - Path=/api/v1/auth/**,/api/users/**.

      - id: RESOURCE-SERVICE
        uri:
           lb://RESOURCE-SERVICE
        predicates:
        - Path=/api/resources/**
```

  4) In our Project Route matching applied on configuration yaml file (named as application.yaml).

     Spring cloud gateway provides a library for building API gateway on top of Spring and java.

- **Eureka Service registry**
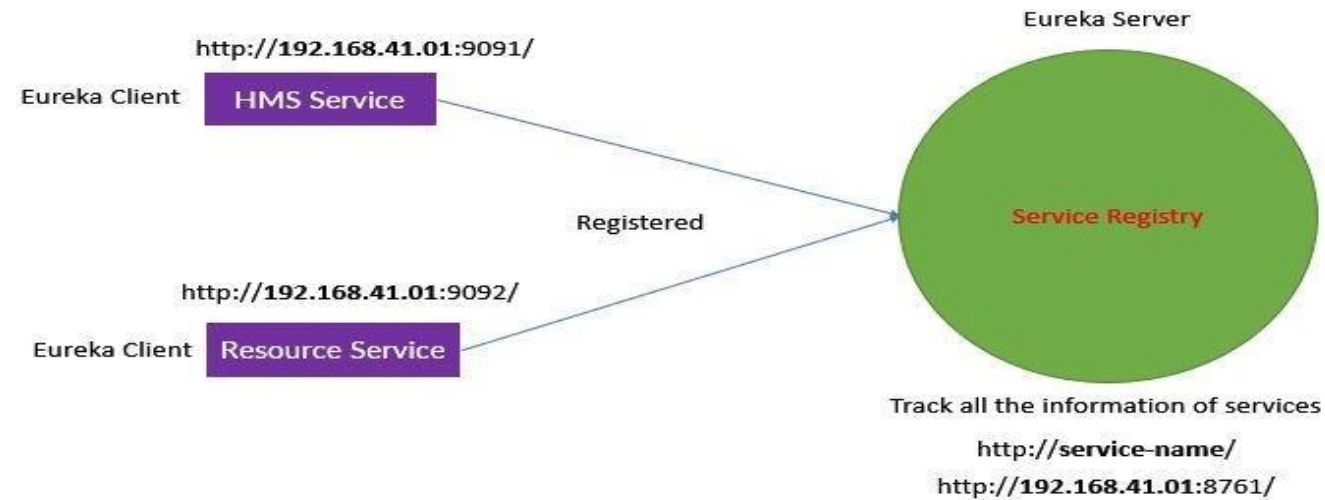
1) Service registry shows that service is Up or down
2) Service registry is used to identify services by name given to them in application.yaml
3) Service Discovery is nothing but a way of locating a service and
Load Balancing is about deciding which service instance to be invoked in case of multiple instances.

- Eureka Server
1) Eureka Server is an application that holds the information about all client-service applications. Every Micro service will register into the Eureka server and Eureka server knows all the client applications running on each port and IP address. Eureka Server is also known as Discovery Server.

2) After downloading the project in main Spring Boot Application class file, we need to add @EnableEurekaServer annotation. The @EnableEurekaServer annotation is used to make your Spring Boot application acts as a Eureka Server.

CONCLUSION

➢ This system aims to simplify the task of the patient and the doctor. It will make patients more relaxed as they do not have to stand in a long queue to fix their appointment and also book an appointment according to their choice in a more convenient way.

➢ Doctors need not worry about managing their appointment. Though you are not going to clinic for taking an appointment, your appointment gets booked from anywhere and however you want.

➢ This helps to save the time of patient. Also the patient can get the doctor of his choice through various filters used in the application. The doctor is also able to view his day to day appointment list which makes it easier for him.

➢ This application will help to optimize the work of patient and doctor. The report above describes how this process has been successfully implemented in the proposed system.

➢ The proposed system has advantages over existing techniques as they lack security mechanism between the hospital and doctor.

➢ Applying such security mechanisms in the communication channel helps to avoid the middle man attack between the

➢ 1.doctor and patient.

➢ 2.doctor and receptionist

➢ 3.patient and receptionist etc…

FUTURE ENHANCEMENT

➢ In Future work, This application to develop based on microservices architecture using different technology etc.

➢ In adding the more features of hospital management system to develop access with user's flexibility.

➢ Addition of more technologies like Jenkins, GraphQl etc.